

MERCURY

A Tool for Slow, Fast, and Hard Real Time Hybrid Simulation

NEES@Colorado



Mercury is a streamlined finite-element simulation tool designed to perform pseudo-dynamic, slow, fast and hard real-time hybrid simulation (for which it is optimized).

Two versions of the code will coexist: a development version in Object Oriented Matlab, and a release version in highly optimized C++ which will operate in: a) Linux with a real time kernel, or b) embedded as a module to run with xPC/Simulink or Real Time LabVIEW. Both will accept the same input file.

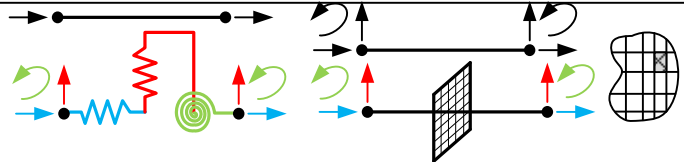
Mercury interfaces directly with SCRAMNET (if necessary). Multithreaded matrix factorization is performed using PARDISO (Intel Math Kernel Library) and element-wise computation is partitioned using a task-based scheduler (Intel Threading Building Blocks).

Mercury can easily be used by non-NEES laboratory for hybrid simulations.

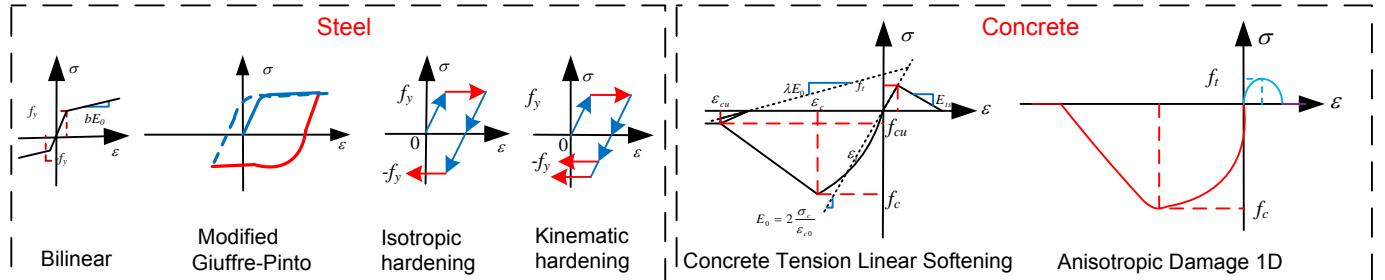
It is anticipated that the Matlab version will play a dominant pedagogical role in familiarizing students/engineers with some of the intricacies of the finite element models implemented in Mercury and other programs.

Element Library

2D Truss; Stiffness and flexibility based Beam Column (with and without element iteration); Zero length element and zero length sections; Fiber sections; New elements/constitutive models could be easily added.



Constitutive Models



Nonlinear Algorithms

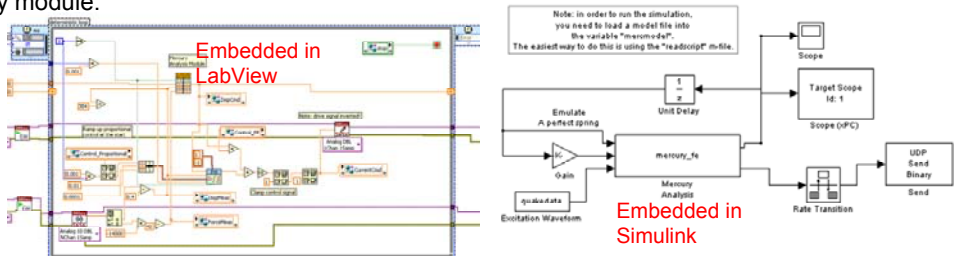
Static Analysis: Newton-Raphson, Modified Newton Raphson; Initial Stiffness; **Transient Analysis:** Newmark β , Hilber-Hughes-Taylor; Shing (initial Stiffness); Kang-Saouma (a new algorithm being developed to accelerate nonlinear time history analysis for RTHS in a modern computational environment).

Multi-Platform Capabilities

Many laboratories use Simulink and/or LabView to interact with instrumentation and hardware. Mercury has the capability to be embedded as a module within both of them and could be wired up to various components depending on the lab setup. Such components include not only actuators, but also shake tables and servomotors. Numerical models of components can also be added by wiring them up to the Mercury module.

Mercury is not limited to RTHS, it can be used for "pseudo-dynamic" tests in a lab simply equipped with either LabView or Simulink.

Mercury can be easily connected to OpenFresco/SIMCOR for distributed testing.



Notes:

- ✓ Mercury is developed through the financial support of the State of Colorado and NEESinc. The Concrete Anisotropic Damage model is developed by the LMT/ENS/Cachan (France).
- ✓ Current status (April 2009): Matlab version 90% completed; c++ version 40% completed.
- ✓ Draft of a ~ 250 pages Theory/User/Validation manual available.
- ✓ Weekly regression testing is performed to verify correctness of the C++ and MATLAB analysis.
- ✓ Will soon be tested in a (non-NEES) real time hybrid simulation of a R/C frame (<http://fhnt.colorado.edu>).
- ✓ For further information, please contact saouma@colorado.edu